**Cloud Native – CaseStudy project Task Screen** **Organic store**– ilan

• Azure CosmoDB API for Mongo Account with Documents - DONE

• Azure Storage Account with blob storage with images uploaded, Queue Storage - DONE

• Azure Functions as per the requirement - Azure http Function to get details from Product table done and Azure http function for uploading Blob image storage.

Az

• Logic Apps as per the requirement – done

• Azure Communication Service configured to facilitate SMS delivery – SMS is available only for US subscriber , however completed Email function

• API Endpoints managed through API Management Service - DONE

• AKS Cluster with Node JS API deployed - DONE

• Azure App Service hosting E Portal -- HOSTED in STATIC WEB APP

Also hosted in Azure app service(Docker and ACS)

• Azure Devops Repos and Pipelines - DONE

AzureCosmoDB API for mongodb

Graphical user interface, text, application, email

Description automatically generated

Execute – node index.js

Postman testing – local host

Graphical user interface, text, application, email

Description automatically generated

**Angular app local run - localhost**

Graphical user interface, website

Description automatically generated

Azure repos

*Git add .*

*Git commit –m first*

*Git push -u origin master*

A screenshot of a computer

Description automatically generated

Agent Pool-- nohup ./run.sh &

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Docker daemon failed.

azureuser@LinuxVM1:~/myagent$ sudo usermod -aG docker $USER

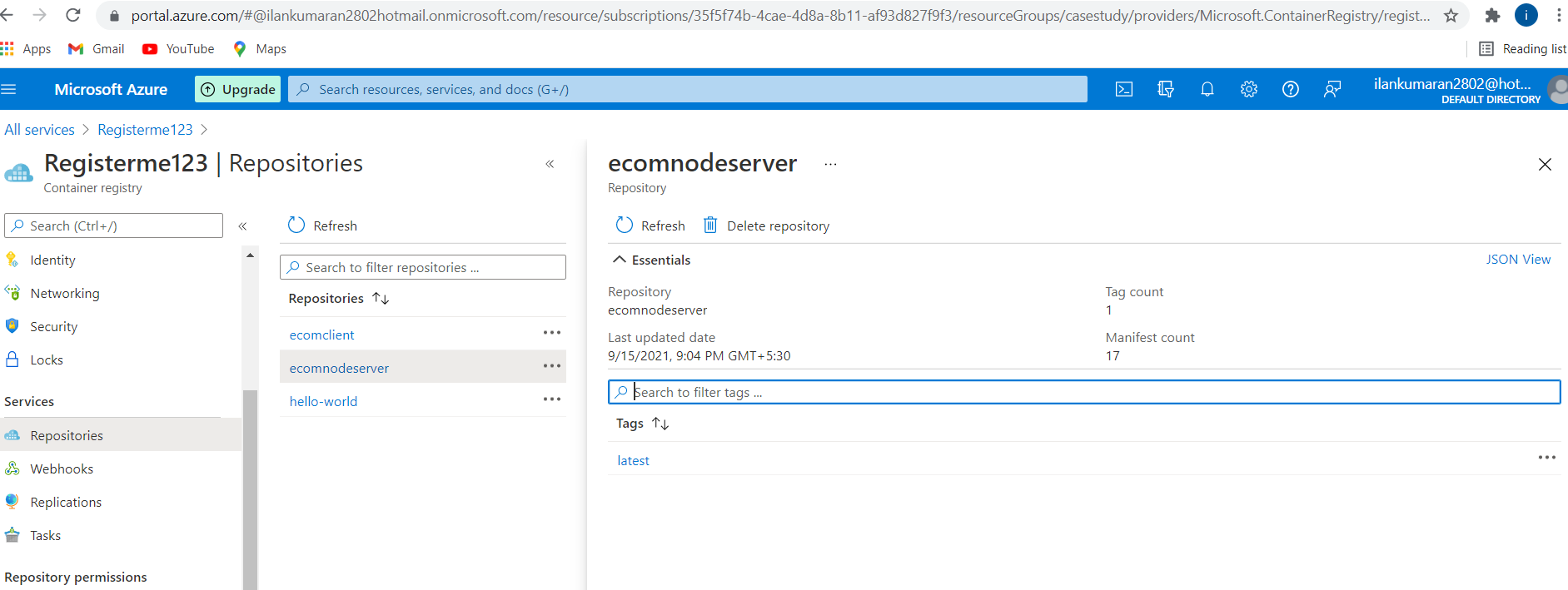
azureuser@LinuxVM1:~/myagent$ sudo chown $USER:docker /var/run/docker.sock

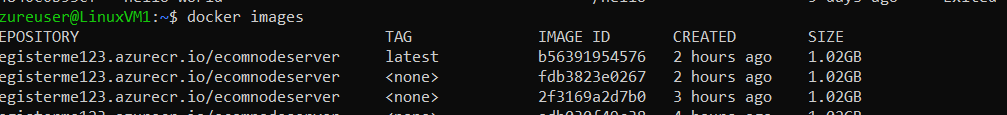
azure repo CI CD NodeJS pipeline

Graphical user interface, application

Description automatically generated

**Docker registry – containerise app**





AkS services - kubernetes

Graphical user interface, text, application, email

Description automatically generated

Kubectl get services – with external ip url

A picture containing calendar

Description automatically generated

Static web app deploy

Graphical user interface, application, email

Description automatically generated

YAML to deploy in static web deploy

trigger:

- master

pool:

  name: ownpool

  vmImage: linuxagent

steps:

- task: NodeTool@0

  inputs:

    versionSpec: '14.x'

  displayName: 'Install Node.js'

- script: |

    npm install -g @angular/cli

    npm install

    ng build --prod

  displayName: 'npm install and build'

- task: AzureStaticWebApp@0

  inputs:

      app\_location: "/"

      api\_location: "api"

      output\_location: "dist/organic-client"

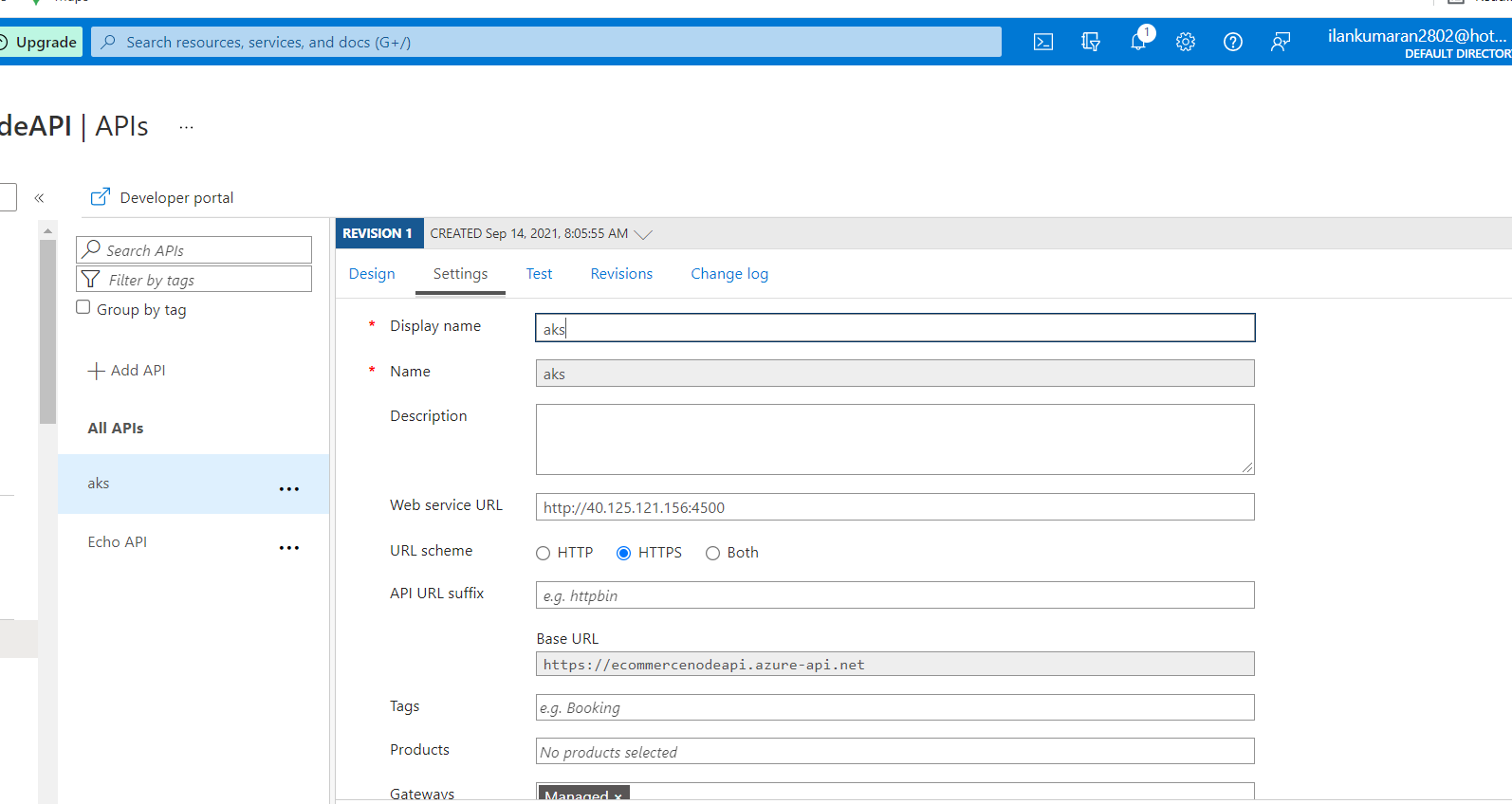
  env:      azure\_static\_web\_apps\_api\_token: $(deployment\_token1)

AZURE BLOB STORAGE :

Graphical user interface, text, application, email

Description automatically generated

AZURE API MANAGEMENT



AKS URl -http://40.125.121.156:4500

API BASE URl :// https://ecommercenodeapi.azure-api.net

Graphical user interface, application

Description automatically generated

AKS service as API management – post man test

Graphical user interface, text, application, email

Description automatically generated

Angular UI – static web running app--- Graphical user interface, application

Description automatically generated

[OrganicClient (brave-dune-054c32b10.azurestaticapps.net)](https://brave-dune-054c32b10.azurestaticapps.net/login)

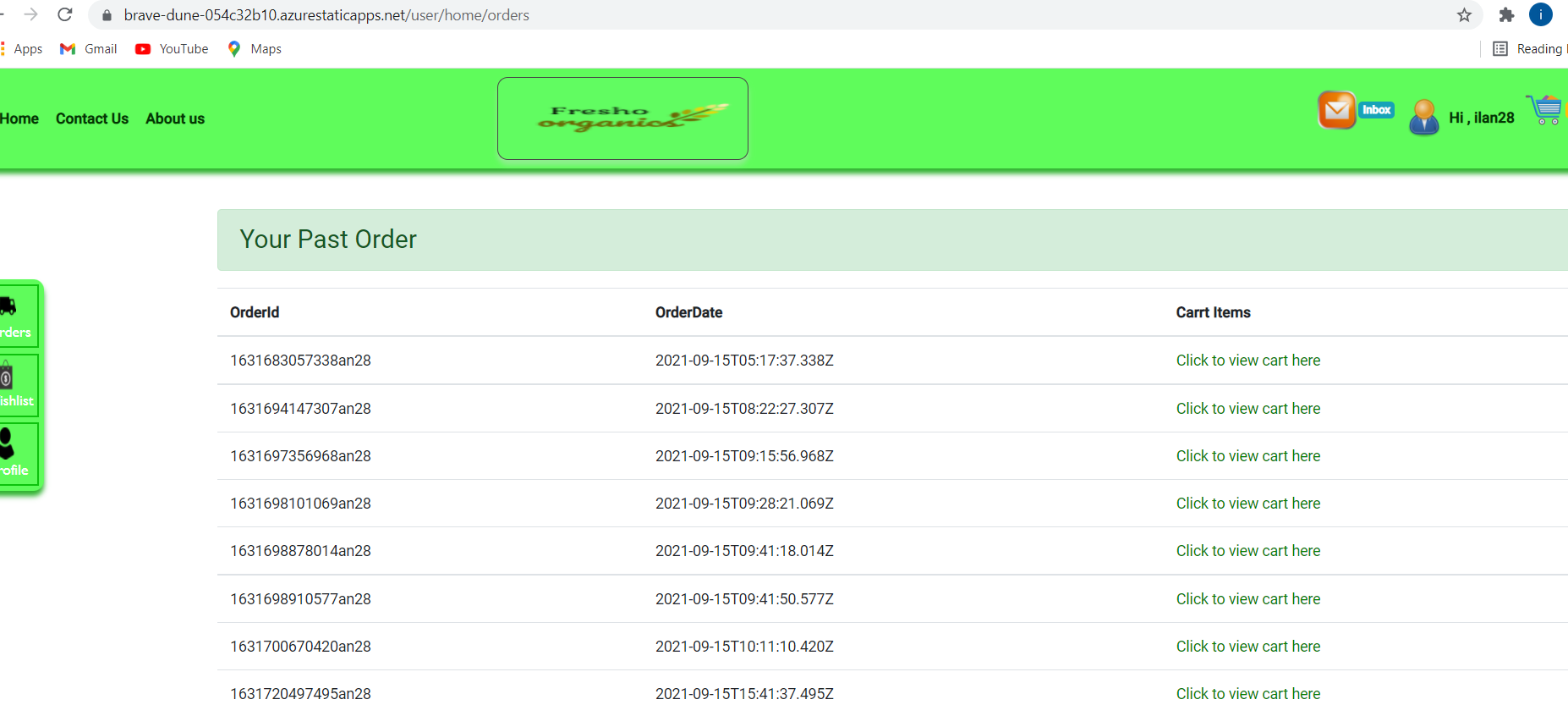
*Ilan28*

*pass@123*

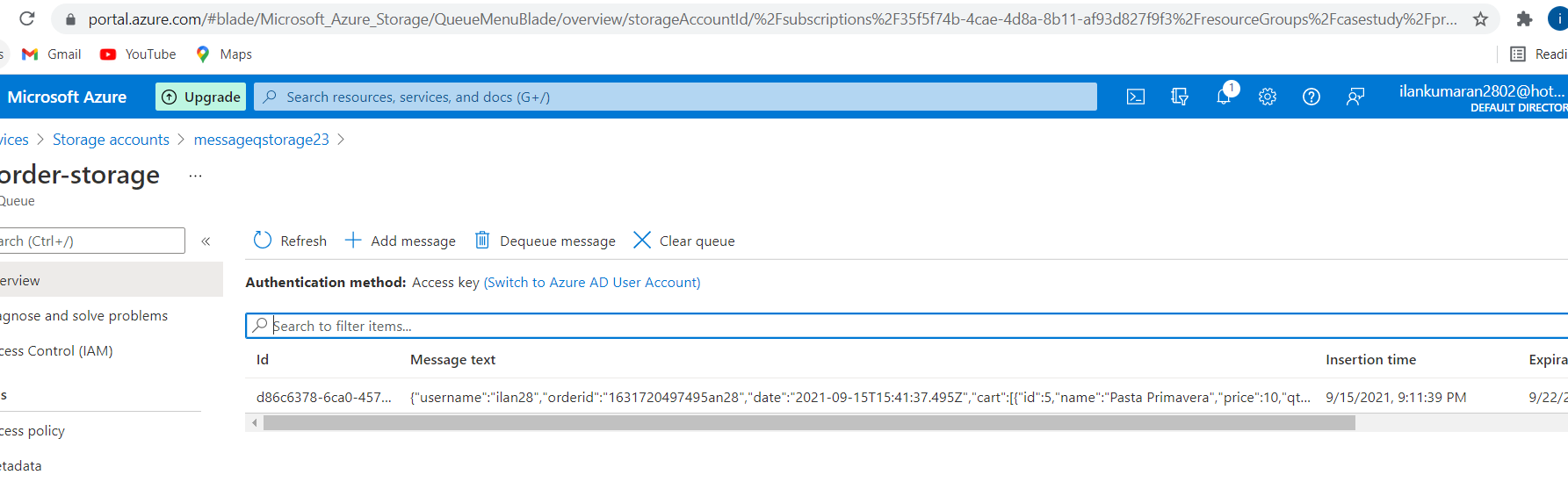
Graphical user interface

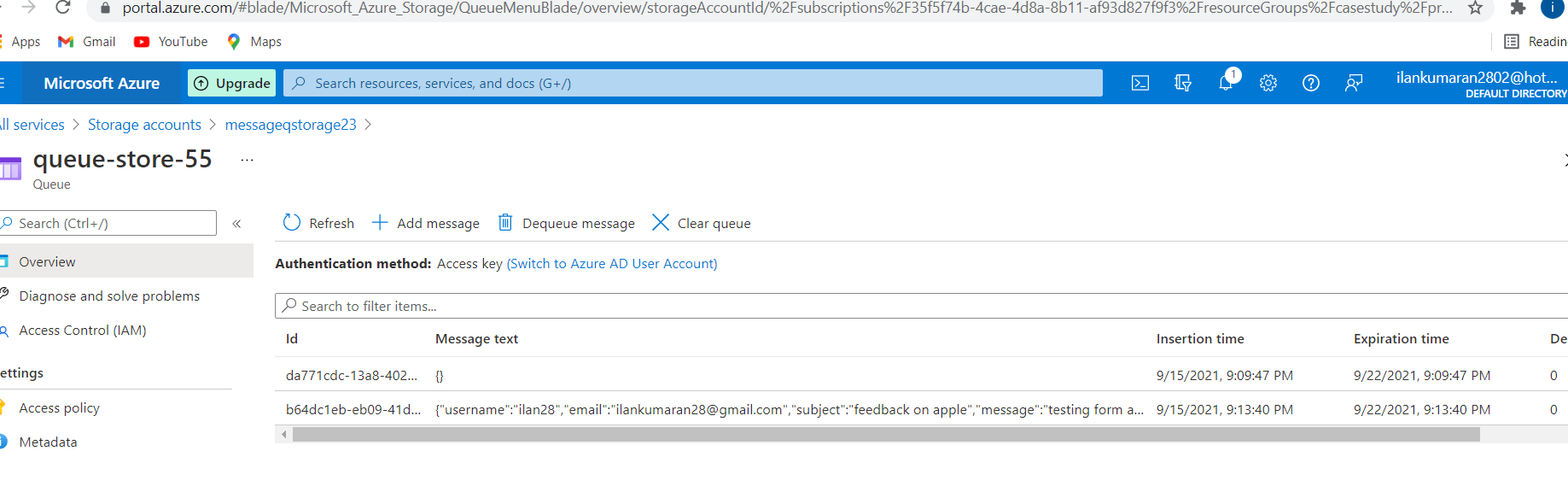
Description automatically generated with medium confidence

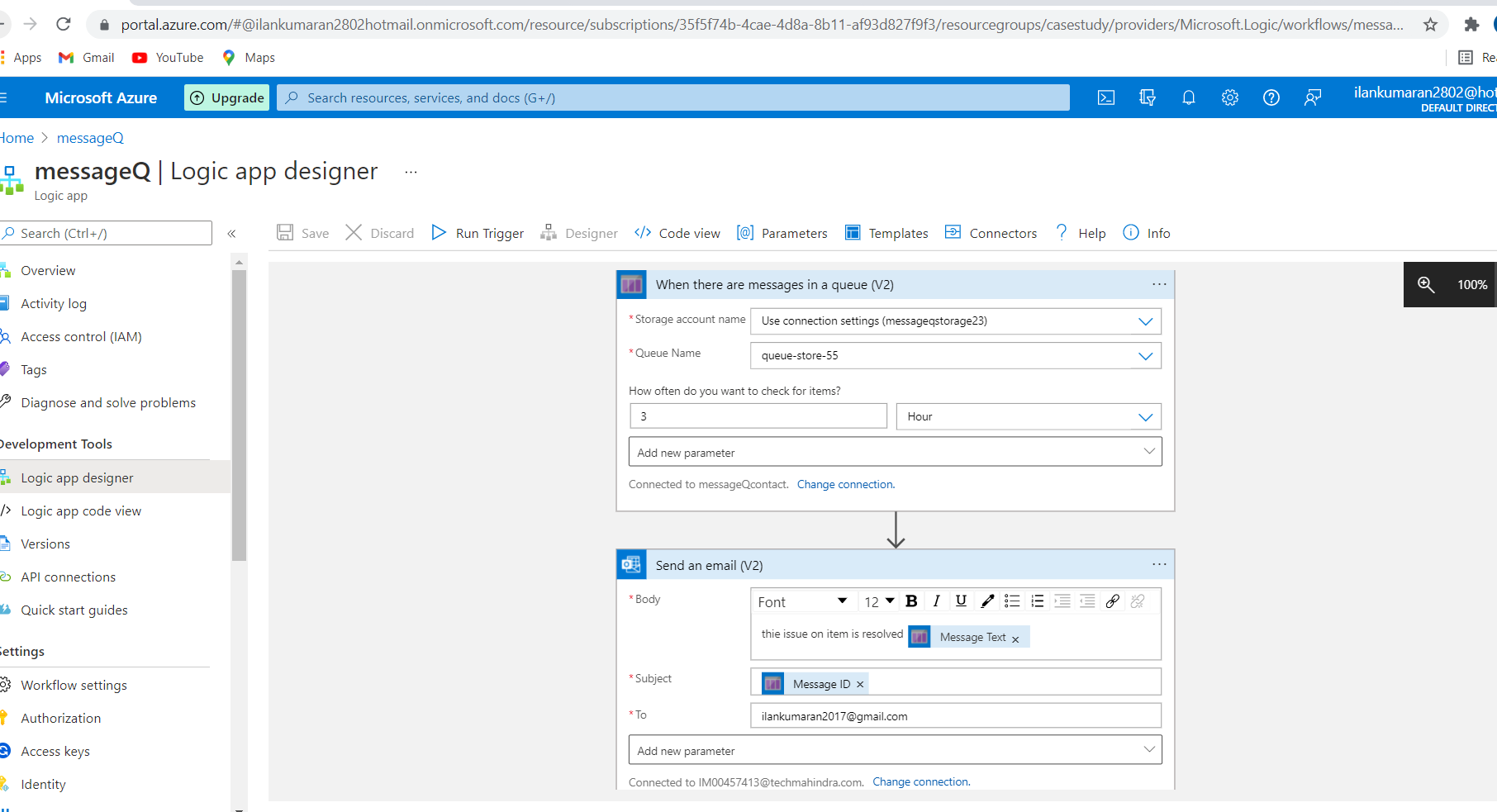
Orders placed.



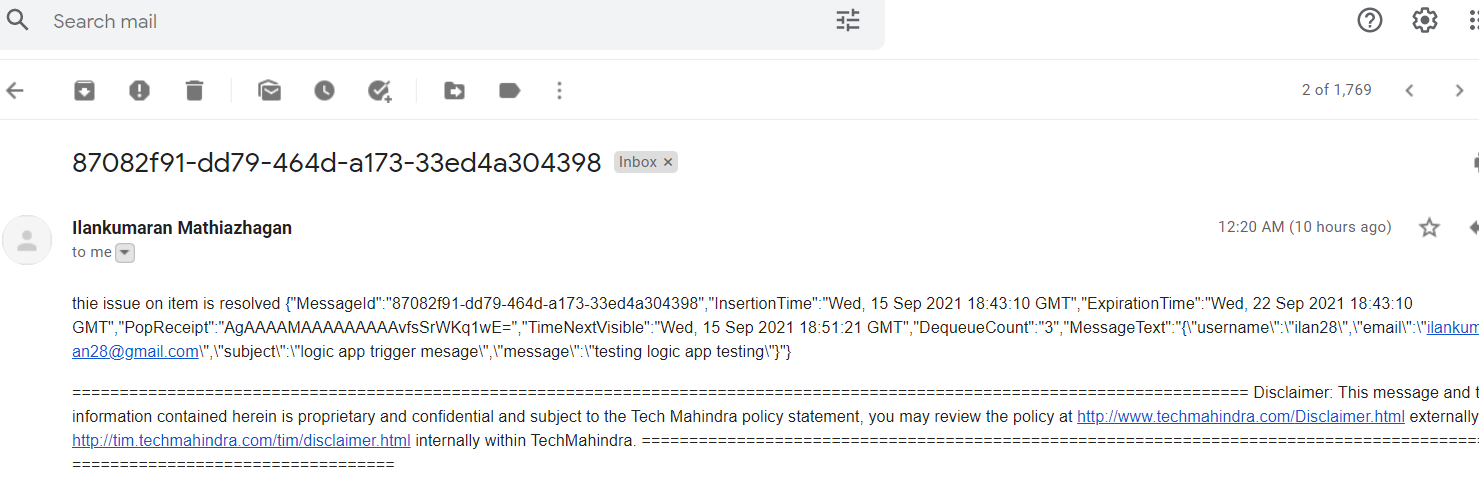
**Storage Queue trigger :**



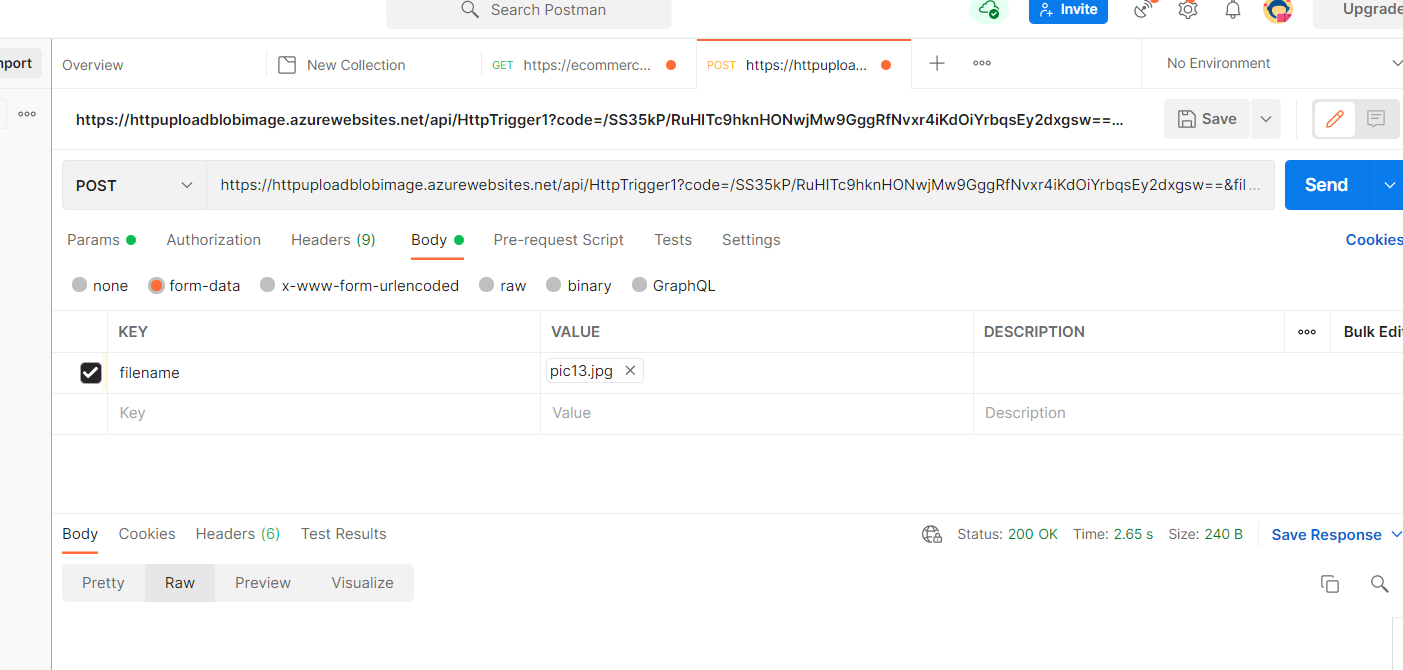


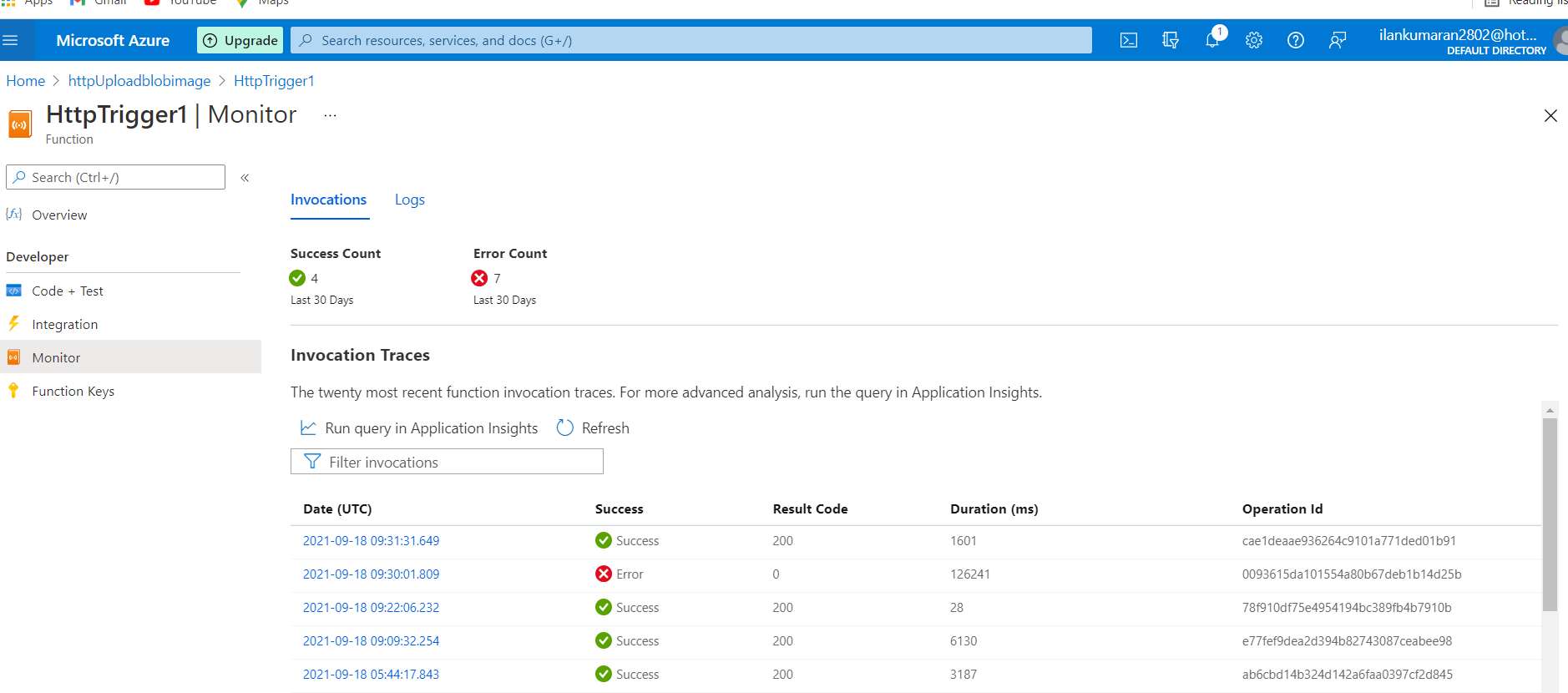
LOGIC APP – send email 

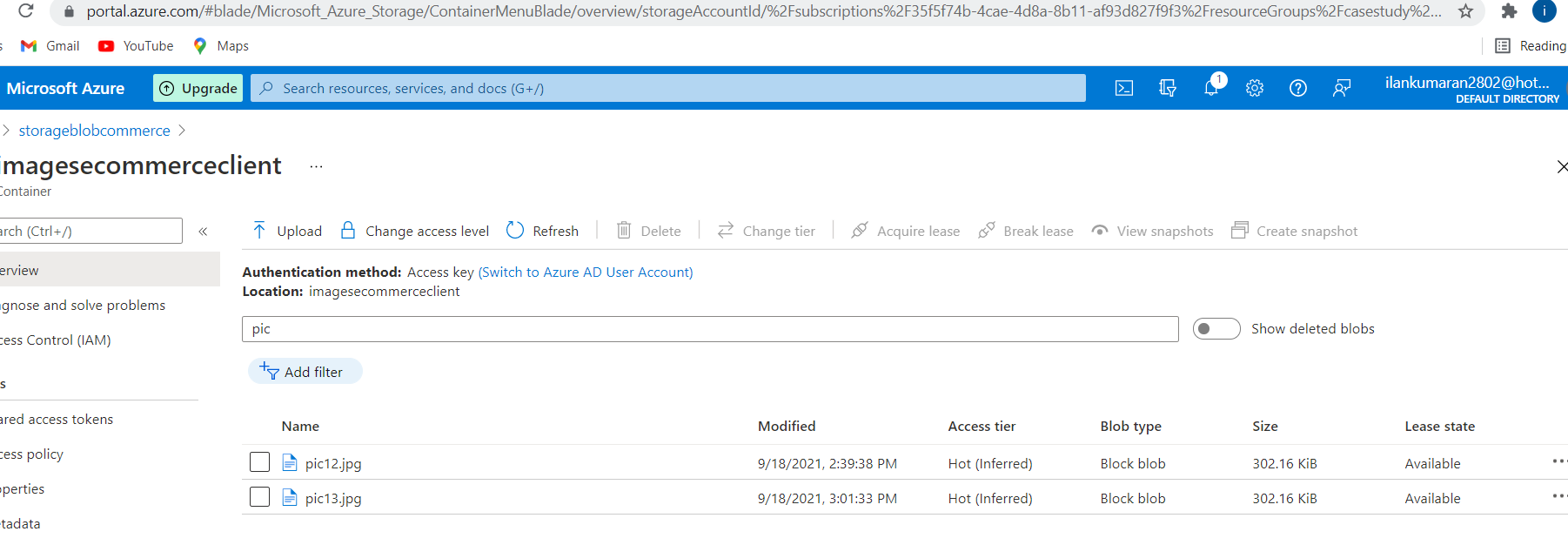
Email received

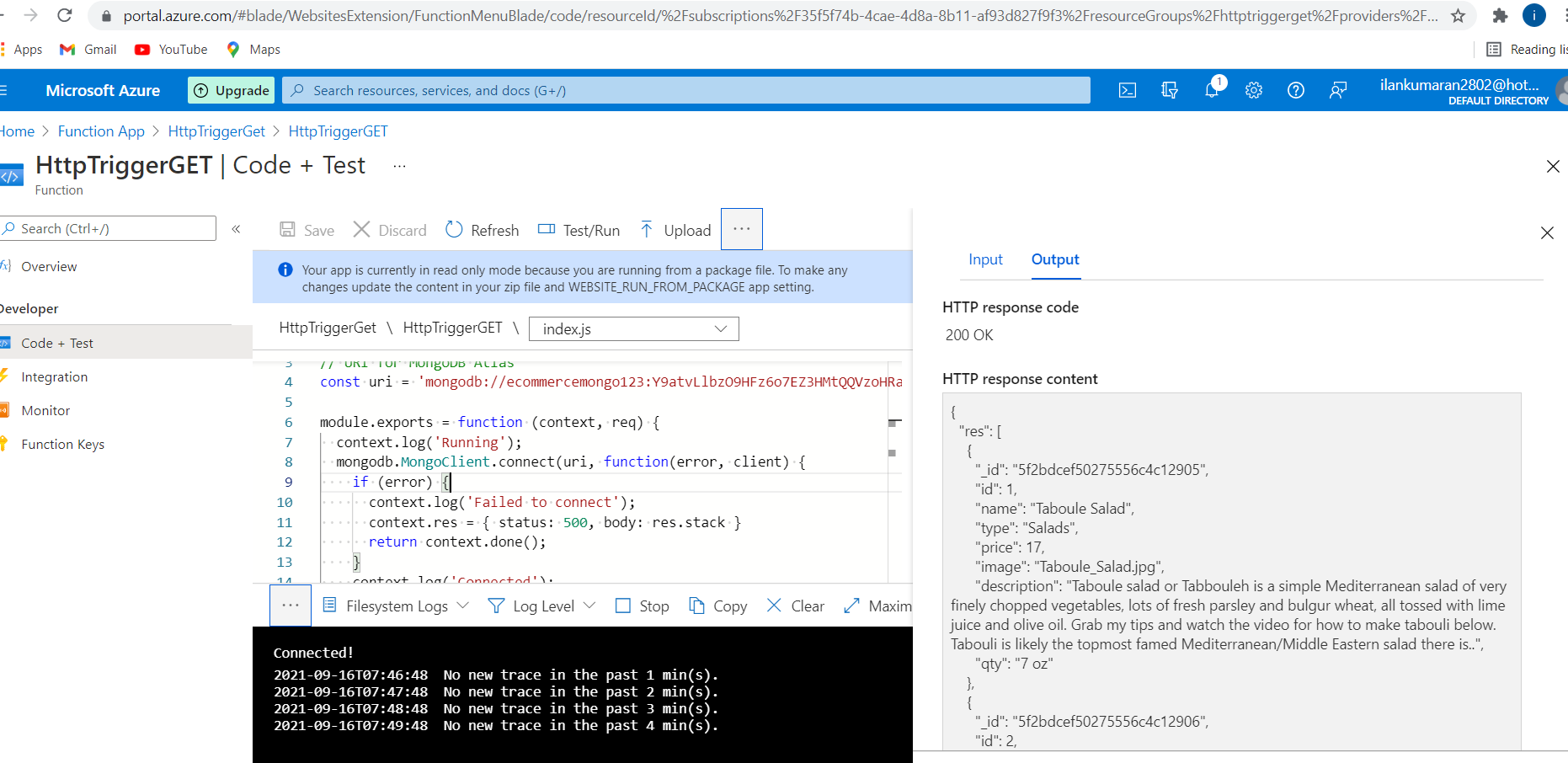


Azure HTTP Function to upload image to blob storage.



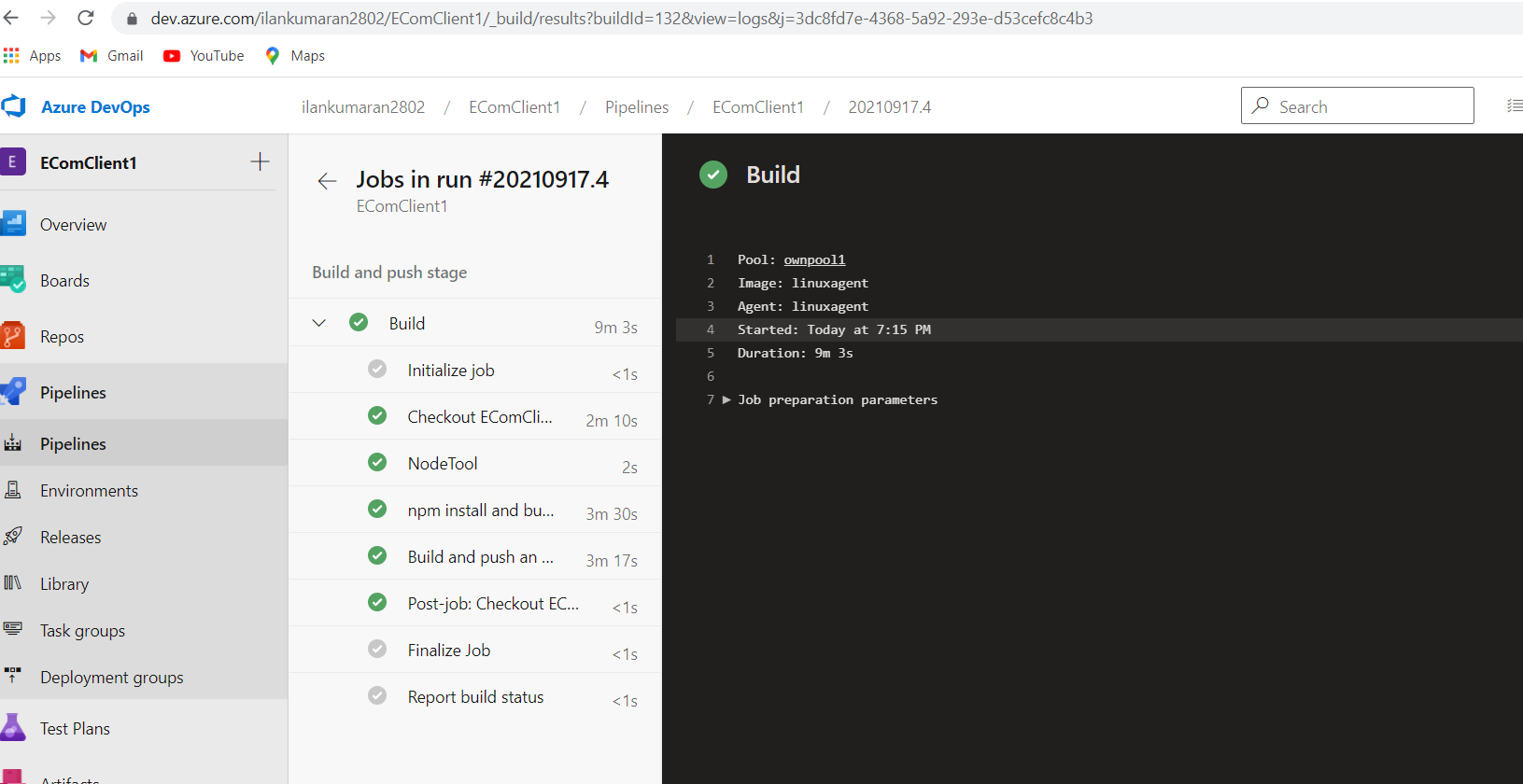




HTTP Trigger azure function – Fetch details from database product table -json 

**2.Deployment approach – organic angular client app using CI/CD docker/ACS and Azure app service**

**New repo for EcomClient1- CI/CD**



Pipeline.yaml

# Docker

# Build and push an image to Azure Container Registry

# https://docs.microsoft.com/azure/devops/pipelines/languages/docker

trigger:

- master

resources:

- repo: self

variables:

  # Container registry service connection established during pipeline creation

  dockerRegistryServiceConnection: 'ef01fc43-f406-4f45-ad1e-e3c35536afc8'

  imageRepository: 'ecomclient1'

  containerRegistry: 'registerme123.azurecr.io'

  dockerfilePath: '$(Build.SourcesDirectory)/Dockerfile'

  tag: 'latest'

  # Agent VM image name

  vmImageName: 'linuxagent'

stages:

- stage: Build

  displayName: Build and push stage

  jobs:

  - job: Build

    displayName: Build

    pool:

      name: ownpool1

      vmImage: $(vmImageName)

    steps:

    - task: NodeTool@0

      inputs:

        versionSpec: '14.x'

        checkLatest: true

    - script: |

          npm install -g @angular/cli

          npm install

          ng build --prod

      displayName: 'npm install and build'

    - task: Docker@2

      displayName: Build and push an image to container registry

      inputs:

              command: buildAndPush

              repository: $(imageRepository)

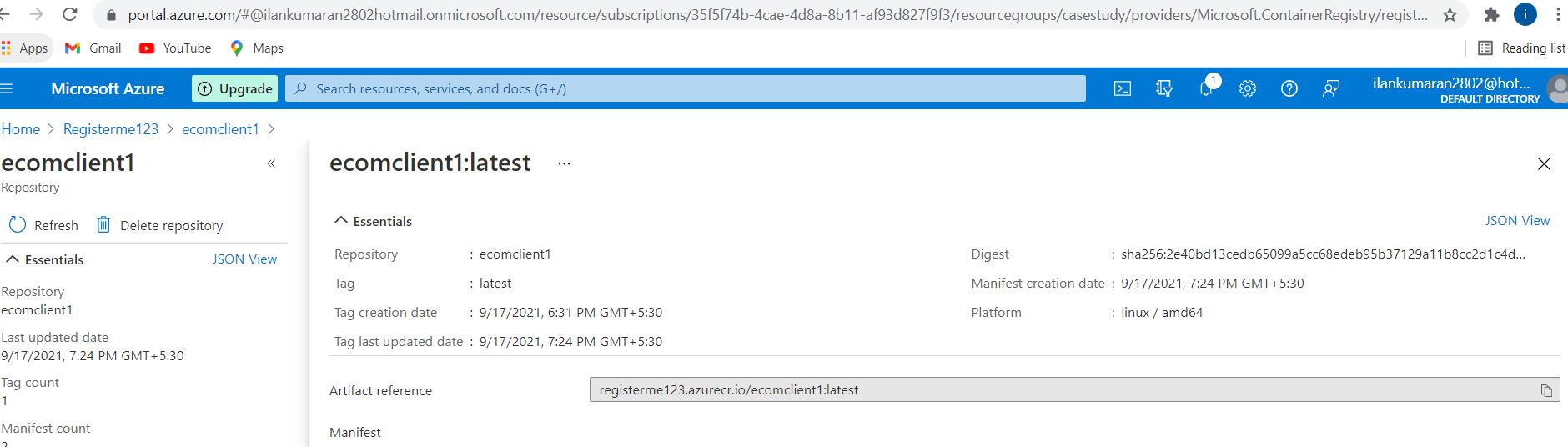
              dockerfile: $(dockerfilePath)

              containerRegistry: $(dockerRegistryServiceConnection)

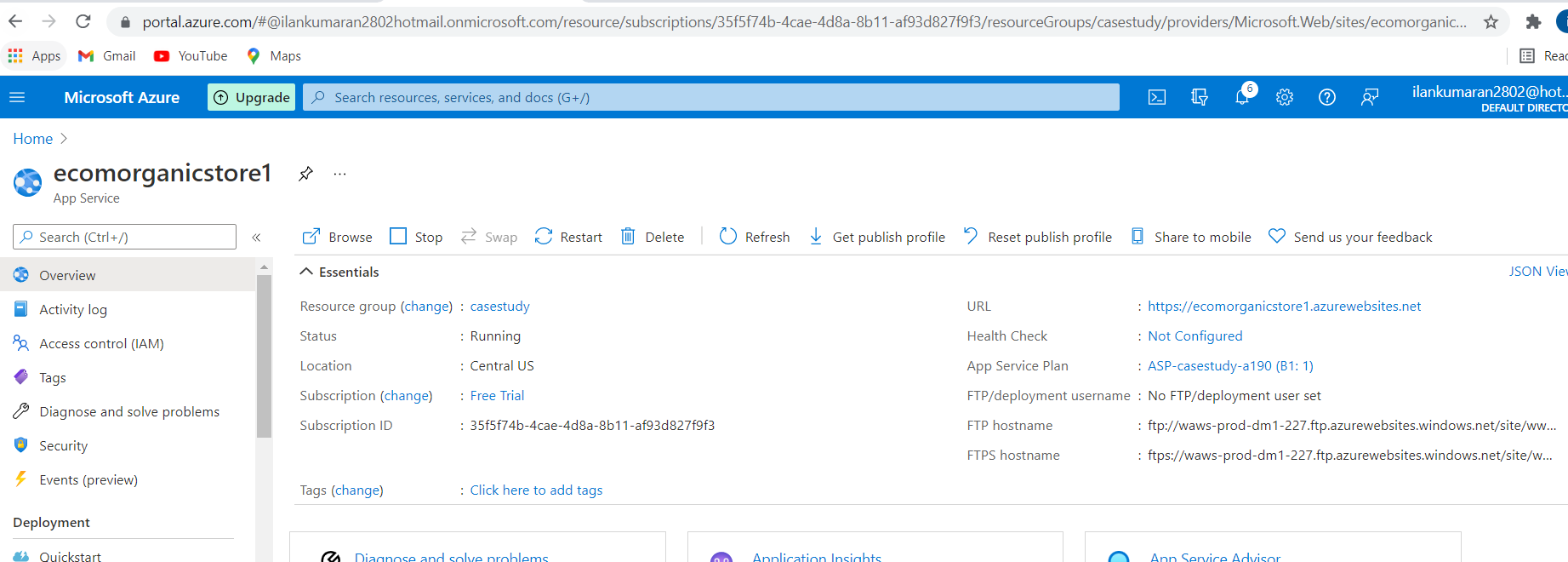
              tags: |

                $(tag)

Azure container service



Azure app service deployed



Live application

<https://ecomorganicstore1.azurewebsites.net/>

